


```
GGGGGGGG LL      BBBB BBBB DDDDDDDD AAAAAA TTTTTTTTTT
GGGGGGGG LL      BBBB BBBB DDDDDDDD AAAAAA TTTTTTTTTT
GG      LL      BB      BB DD      DD AA      AA TT
GG      LL      BB      BB DD      DD AA      AA TT
GG      LL      BB      BB DD      DD AA      AA TT
GG      LL      BBBB BBBB DD      DD AA      AA TT
GG      LL      BBBB BBBB DD      DD AA      AA TT
GG      LL      BB      BB DD      DD AAAAAAAAAA TT
GG      LL      BB      BB DD      DD AAAAAAAAAA TT
GG      LL      BB      BB DD      DD AA      AA TT
GG      LL      BBBB BBBB DDDDDDDD AA      AA TT
GGGGGG LL      BBBB BBBB DDDDDDDD AA      AA TT
GGGGGG LL      BBBB BBBB DDDDDDDD AA      AA TT
GGGGGG LL      LLLLLLLLLL
GGGGGG LL      LLLLLLLLLL
```

```
LL      IIIIII SSSSSSSS
LL      IIIIII SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      IIIIII SSSSSSSS
LLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLL IIIIII SSSSSSSS
```



```
0001 0 %TITLE 'Define global literals and data structures'
0002 0 MODULE glbdat ( IDENT = 'V04-000'
0003 0 ) =
0004 1 BEGIN
0005 1
0006 1 *****
0007 1 *
0008 1 *   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0009 1 *   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0010 1 *   ALL RIGHTS RESERVED.
0011 1 *
0012 1 *   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0013 1 *   ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0014 1 *   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0015 1 *   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0016 1 *   OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0017 1 *   TRANSFERRED.
0018 1 *
0019 1 *   THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0020 1 *   AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0021 1 *   CORPORATION.
0022 1 *
0023 1 *   DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0024 1 *   SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0025 1 *
0026 1 *
0027 1 *****
0028 1
0029 1
0030 1 ++
0031 1 FACILITY:      DSR (Digital Standard RUNOFF) / DSRPLUS
0032 1
0033 1 ABSTRACT:      Global data definitions; contains no instructions.
0034 1
0035 1 ENVIRONMENT:   Transportable
0036 1
0037 1 AUTHOR:       R.W.Friday      CREATION DATE:  November, 1978
0038 1
```

```

40 0039 1 XSBTTL 'Revision History'
41 0040 1   MODIFIED BY:
42 0041 1
43 0042 1
44 0043 1   041 REM00041 Ray Marshall 1-June-1984
45 0044 1   Moved XPD_APPEND back into the SAVE/RESTORE area and
46 0045 1   conditionalized it and XPD_FLAGS for DSRPLUS only. Also
47 0046 1   worked on some of the comments in that area.
48 0047 1
49 0048 1   040 KFA00040 Ken Alden 17-Apr-1984
50 0049 1   For DSRPLUS:
51 0050 1   Added a semicolon to the end of the command that
52 0051 1   just precedes 'OWN'.
53 0052 1
54 0053 1   039 REM00039 Ray Marshall 16-Feb -> 15-Mar-1984
55 0054 1   Added one literal, one global variable, and two global bind
56 0055 1   structures containing filetype lists used in mapping the
57 0056 1   output filetype based on the input filetype. These items
58 0057 1   are also used when the user defaults the input filetype.
59 0058 1
60 0059 1   The literal is IPFTCT and indicates how many cells there are
61 0060 1   in the list of input filetypes in IPFTOP. There are also as
62 0061 1   many cells in the list of output filetypes in OPFTOP. The
63 0062 1   variable is IPFTYP and will contain the index into the above
64 0063 1   mentioned filetype lists.
65 0064 1
66 0065 1   Also added LFFCTX which is a global context cell used by the
67 0066 1   VMS RTL routine LIB$FIND_FILE.
68 0067 1
69 0068 1   038 REM00038 Ray Marshall 17-November-1983
70 0069 1   Modified the external definition of ATABLE to use the new
71 0070 1   macro ATABLE_DEFINITION defined in ATCODE.REQ.
72 0071 1
73 0072 1   037 REM00037 Ray Marshall 7-November-1983
74 0073 1   Doubled the size of ATABLE (added another 128 cells) to support
75 0074 1   the DEC multinational character set. Also, changed it to a
76 0075 1   BYTE VECTOR.
77 0076 1
78 0077 1   036 KFA00036 Ken Alden 30-Sep-1983
79 0078 1   Moved items around for save/restore fix.
80 0079 1
81 0080 1   035 REM00035 Ray Marshall 22-Jul-1983
82 0081 1   Conditionalized TEMP_SIZE_MRA for DSRPLUS only.
83 0082 1
84 0083 1   034 REM00034 Ray Marshall 22-Jul-1983
85 0084 1   Changed TEMP MRA SIZE to TEMP SIZE MRA so that it will be
86 0085 1   unique within the first 6 characters for the TOPS-20 linker.
87 0086 1   Added KWTCBR for future addition of cross-book cross-reference
88 0087 1   support.
89 0088 1
90 0089 1   033 RER00033 Ron Randall 9-Jun-1983
91 0090 1   For DSRPLUS:
92 0091 1   Added flag to indicate when GETNUM is processing an entity.
93 0092 1
94 0093 1   032 RER00032 Ron Randall 11-May-1983
95 0094 1   For DSRPLUS:
96 0095 1   Added general-purpose temporary and hold mra and tsf.
```


97	0096	1	031	KAD00031	Keith Dawson	9-May-1983	
98	0097	1		Remove support for .DX, .PX.			
99	0098	1					
100	0099	1	030	KFA00030	Ken Alden	14-Apr-1983	
101	0100	1		Alter headerlevel, pagesize to hyphenated strings.			
102	0101	1					
103	0102	1	029	KAD00029	Keith Dawson	13-Apr-1983	
104	0103	1		Added omitted DSRPLUS conditional on \$keyword_tables.			
105	0104	1					
106	0105	1	028	RER00028	Ron Randall	06-Apr-1983	
107	0106	1		For DSRPLUS:			
108	0107	1		Added secondary topnote IOBs and footnote numbering variables.			
109	0108	1					
110	0109	1	027	REM00027	Ray Marshall	06-Apr-1983	
111	0110	1		Fix misspelled literal in a keyword table definition.			
112	0111	1					
113	0112	1	026	REM00023	Ray Marshall	Mar-1983	
114	0113	1		Added KWTRPN, KWTUDM, KWTPDM, & KWTPDP keyword table listheads.			
115	0114	1		Also built the predefined keyword table entries.			
116	0115	1					
117	0116	1	025	RER00025	Ron Randall	20-Mar-1983	
118	0117	1		Fine tuning of topnote globals.			
119	0118	1		Deleted passthrough flag.			
120	0119	1					
121	0120	1	024	KAD00024	Keith Dawson	20-Mar-1983	
122	0121	1		Removed LN01 conditionals and all references to .BIX			
123	0122	1		and .BTC files.			
124	0123	1					
125	0124	1	023	RER00023	Ron Randall	19-Mar-1983	
126	0125	1		For DSRPLUS: Added definitions related to topnotes.			
127	0126	1					
128	0127	1	022	KFA00022	Ken Alden	18-Mar-1983	
129	0128	1		Changes made were to make SAVE/RESTORE related globals			
130	0129	1		visible to DSR.			
131	0130	1					
132	0131	1	021	KAD00021	Keith Dawson	07-Mar-1983	
133	0132	1		Global edit of all modules. Updated module names, idents,			
134	0133	1		copyright dates. Changed require files to BLISS library.			
135	0134	1					
136	0135	1					

```
: 138 0136 1 %SBTTL 'Require files and libraries'
: 139 0137 1
: 140 0138 1 INCLUDE FILES:
: 141 0139 1
: 142 0140 1 LIBRARY 'NXPORT:XPORT';           ! XPORT Library
: 143 0141 1 REQUIRE 'REQ:RNODEF';           ! RUNOFF variant definitions
: 144 0272 1
: 145 U 0273 1 %IF DSRPLUS %THEN
: 146 U 0274 1 LIBRARY 'REQ:DPLLIB';           ! DSRPLUS BLISS Library
: 147 0275 1 %ELSE
: 148 0276 1 LIBRARY 'REQ:DSRLIB';           ! DSR BLISS Library
: 149 0277 1 %FI
: 150 0278 1
: 151 0279 1
: 152 0280 1 EQUATED SYMBOLS:
: 153 0281 1
: 154 0282 1 GLOBAL LITERAL
: 155 0283 1 rintex = %0'34' : UNSIGNED (8), ! Runoff internal escape character.
: 156 0284 1
: 157 0285 1 The global literal s_fmra is used in the module FNONLY. In that
: 158 0286 1 module, it is necessary to reset the MRA size explicitly. The
: 159 0287 1 global literal assures that the size will be reset to the allocated
: 160 0288 1 value.
: 161 0289 1
: 162 0290 1 s_fmra = 500,
: 163 0291 1
: 164 0292 1 tnmra_size is similarly used in the module TPNOTE.
: 165 0293 1
: 166 0294 1 tnmra_size = 500,
: 167 0295 1
: 168 U 0296 1 %IF DSRPLUS %THEN
: 169 U 0297 1
: 170 U 0298 1 temp_size_mra is similarly used in any module.
: 171 U 0299 1
: 172 U 0300 1 temp_size_mra = 100,
: 173 0301 1 %FI
: 174 U 0302 1 ipftct = %IF dsrplus %THEN 14;   ! Number of input file types known.
: 175 0303 1 %ELSE 12;                         ! If IPFTOP & OPFTOP are changed,
: 176 0304 1 %FI                               ! this literal may also need changing.
: 177 0305 1
: 178 0306 1 OWN STORAGE:
: 179 0307 1
: 180 0308 1 OWN
: 181 0309 1 pp_sca : $h_r_sca_block;           ! Used in PUSH_SCA, POP_SCA macros
: 182 0310 1
```



```
184 0311 1 %SBTTL 'Global storage'
185 0312 1
186 0313 1 GLOBAL STORAGE:
187 0314 1
188 0315 1 GLOBAL
189 0316 1 r_cmd : $rno_cmd; ! Command-line information.
190 0317 1
191 0318 1 GLOBAL
192 0319 1 lststf : COUNTED_LIST,
193 0320 1 tsfa : VECTOR [tsf_size],
194 0321 1 fs_allocate (mraa, 500),
195 0322 1 fs_allocate (fra, 500),
196 0323 1 fs_allocate (ira, 10),
197 0324 1 fs_allocate (xmra, 100), ! Used for <INDEX flag> processing.
198 0325 1 mra, ! Refers to MRAa or MRAb.
199 0326 1
200 0327 1 Descriptors for title and subtitle.
201 0328 1
202 0329 1 tittsf : VECTOR [tsf_size], ! Titles
203 0330 1 sbttsf : VECTOR [tsf_size], ! Subtitles
204 0331 1
205 0332 1 Storage for titles and subtitles
206 0333 1
207 0334 1 fs_allocate (titmra, max_title), ! Titles
208 0335 1 fs_allocate (sbtmra, max_title), ! Subtitles
209 0336 1
210 0337 1 xtsf : VECTOR [tsf_size], ! Used for <INDEX flag> processing.
211 0338 1 khar, ! Character storage
212 0339 1
213 0340 1 + This is the beginning of the SAVE/RESTORE area *****
214 0341 1
215 0342 1 ANY changes in this area should be reflected in SAVE.REQ as well.
216 0343 1 -
217 0344 1 flgt : flag_table [flag_count],
218 0345 1 ttable : COUNTED_LIST [max_tabs], ! Tab settings
219 0346 1
220 0347 1 The following definition macro is defined in ATCODE.REQ to be [256, BYTE].
221 0348 1
222 0349 1 atable : atable_definition, ! Action table. Used to identify
223 0350 1 ! what type of action is to be taken
224 0351 1 ! on encountering any given character.
225 0352 1 pdtstf : VECTOR [pdt_size],
226 0353 1 save : $save_block, ! This is the area where single
227 0354 1 ! items are stored.
228 0355 1
229 0356 1 + This marks the end of the SAVE block.
230 0357 1 Below is the additional area for SAVE ALL.
231 0358 1 -
232 0359 1 saveall : $saveall_block, ! This area holds more items saved
233 0360 1 ! only during a .SAVE ALL
234 0361 1 dattim : VECTOR [6], ! Day and time pieces.
235 0362 1 hllist : COUNTED_LIST [max_levels],
236 0363 1 ecc : $ecc_blockvector, ! Entity counter / caption area.
237 0364 1 %IF DSRPLUS %THEN
238 0365 1
239 0366 1 ! These DSRPLUS structures are updated by setting the default
240 0367 1 ! XPLUS attributes, .ENABLE XPLUS (attr,...).
```

```
241 U 0368 1 !
242 UU 0369 1 xpd_flags, ! Default attributes flags.
243 U 0370 1 fs_allocate (xpd_append, xp_string_size), ! Default APPEND = string.
244 0371 1 %FI
245 0372 1 !+
246 0373 1 ! This is the end of the SAVE/RESTORE area *****
247 0374 1 !-
248 0375 1 savstk : savstack, ! Storage for the save/restore list.
249 0376 1
250 U 0377 1 %IF DSRPLUS %THEN
251 UU 0378 1 ! Topnote related globals.
252 UU 0379 1
253 UU 0380 1
254 UU 0381 1 topnot : tn_definition, ! Topnote control table.
255 UU 0382 1 tnrec : tnrec_definition, ! Topnote file record control area.
256 UU 0383 1 tniob : REF $XPO_IOB (), ! Internal topnote IOB.
257 UU 0384 1 tnuob : REF $XPO_IOB (), ! User topnote IOB.
258 UU 0385 1 tn0iob : $XPO_IOB (),
259 UU 0386 1 tn1iob : $XPO_IOB (),
260 U 0387 1 tnpdt : VECTOR [pdt_size], ! Save area for paragraph stuff.
261 U 0388 1 tnsca : VECTOR [sca_size], ! Save area for SCA.
262 U 0389 1 tntsf : VECTOR [tsf_size], ! Substitute TSF.
263 UU 0390 1
264 UU 0391 1 tnhdh,
265 UU 0392 1 tnhlch,
266 UU 0393 1 tnhlsc,
267 UU 0394 1 tnhlst,
268 UU 0395 1 tnhmra,
269 UU 0396 1 tnhpdt,
270 UU 0397 1 tnhrch,
271 UU 0398 1 tnhtsf,
272 UU 0399 1 fs_allocate (tnmra, tnmra_size),
273 U 0400 1 tn_ldd : VECTOR [3], ! Substitute display descriptor.
274 U 0401 1 tn_lchr : VECTOR [3], ! Substitute .LIST bullet char.
275 U 0402 1 tn_list : COUNTED_LIST [3], ! Substitute .LIST counters
276 U 0403 1 tn_llchr : VECTOR [3], ! Substitute left character
277 U 0404 1 tn_lrchr : VECTOR [3], ! Substitute right character
278 U 0405 1 tn_lskp : VECTOR [3], ! Substitute .LIST el spacing.
279 0406 1
280 U 0407 1 ! Footnote numbering variables.
281 U 0408 1
282 U 0409 1 l_character,
283 U 0410 1 r_character,
284 U 0411 1
285 U 0412 1 ! The following structures are updated each time the
286 UU 0413 1 ! ATTRIB routine is called.
287 U 0414 1
288 U 0415 1 xp_flags, ! Working attributes flags
289 U 0416 1 fs_allocate (xp_append, xp_string_size), ! Working APPEND = string
290 U 0417 1 fs_allocate (xp_sort, xp_string_size), ! Working SORT = string
291 U 0418 1
292 UU 0419 1 ! General-purpose mra and tsf substitution aids.
293 UU 0420 1
294 U 0421 1 hold_mra, ! Holds MRA address.
295 U 0422 1 fs_allocate (temp_mra, temp_size_mra), ! Substitute MRA.
296 U 0423 1 hold_tsf, ! Holds TSF address.
297 U 0424 1 temp_tsf : VECTOR [tsf_size], ! Substitute TSF.
```



```
298 U 0425 1 |
299 U 0426 1 | Flag to indicate when GETNUM is processing a $ or $$ entity.
300 U 0427 1 |
301 U 0428 1 | processing_entity, | Used by NEWSUB.
302 U 0429 1 | entity_in_footnote, | Used by FCIMRA.
303 U 0430 1 | %FI
304 U 0431 1 |
305 U 0432 1 | tsf : REF VECTOR [tsf_size], | Refers to TSFa or TSFb.
306 U 0433 1 | fnesiz : fn_ext_size_definition, | Sizes of pending footnotes.
307 U 0434 1 | fnisiz : fn_int_size_definition, | Number of TSF/MRA pairs in
308 U 0435 1 | | each pending footnote.
309 U 0436 1 | fnct : fnct_definition, | Footnote control table.
310 U 0437 1 | gca : gca_definition, | Global control area.
311 U 0438 1 | hct : hct_definition, | Header control area.
312 U 0439 1 | ifstk : ifstack, | Stack for .IFs, and so on.
313 U 0440 1 | fohlch,
314 U 0441 1 | fohrch,
315 U 0442 1 | fohdd,
316 U 0443 1 | fohlsc,
317 U 0444 1 | fohlsc,
318 U 0445 1 | fohlst,
319 U 0446 1 | fohpdt,
320 U 0447 1 | fohmra,
321 U 0448 1 | fohtsf,
322 U 0449 1 | foorec : foorec_definition,
323 U 0450 1 | fs_allocate (foomra, s_fmra),
324 U 0451 1 |
325 U 0452 1 | NOTE: An RSX IOB gets 80 (120 octal) words allocated to it.
326 U 0453 1 |
327 U 0454 1 | fooiob : REF $XPO_IOB (), | Used for processing footnotes.
328 U 0455 1 | fotiob : REF $XPO_IOB (),
329 U 0456 1 | foliob : $XPO_IOB (), | Never referred to directly
330 U 0457 1 | fo2iob : $XPO_IOB (), | Never referred to directly
331 U 0458 1 | foopdt : VECTOR [pdt_size], | Save area for paragraph stuff.
332 U 0459 1 | foosca : VECTOR [sca_size], | Save area for SCA
333 U 0460 1 | f_list : COUNTED_LIST [3], | Substitute .LIST counters
334 U 0461 1 | f_lchr : VECTOR [3], | Substitute .LIST bullet char.
335 U 0462 1 | f_lskp : VECTOR [3], | Substitute .LIST EL spacing.
336 U 0463 1 | f_llchr : VECTOR [3], | Substitute left character
337 U 0464 1 | f_lrchr : VECTOR [3], | Substitute right character
338 U 0465 1 | f_ldd : VECTOR [3], | Substitute display descriptor.
339 U 0466 1 | footsf : VECTOR [tsf_size], | Substitute TSF.
340 U 0467 1 | irac : irac_definition,
341 U 0468 1 | npagen : page_definition, | Page number of next page
342 U 0469 1 | spager : BLOCK [max_page_ranges * page_sct_size], | List of starting pages
343 U 0470 1 | tpager : BLOCK [max_page_ranges * page_sct_size], | List of terminating pages.
344 U 0471 1 | pagen : page_definition, | Page number of current page
345 U 0472 1 | brnoob : $XPO_IOB (), | IOB for binary index and IOC output.
346 U 0473 1 | brniob : $XPO_IOB (), | IOB for binary crossref input.
347 U 0474 1 | ndxpol : INITIAL (0), | Address of indexing pool.
348 U 0475 1 | ndxsge : INITIAL (0), | End of current segment.
349 U 0476 1 | ndxsgf : INITIAL (0), | No storage in current segment.
350 U 0477 1 | xtncnt, | Number of XTNTAB entries.
351 U 0478 1 | xtnlsp : INITIAL (0),
352 U 0479 1 | xtnlsx : INITIAL (0),
353 U 0480 1 | xtnpol : INITIAL (0),
354 U 0481 1 | xtngsp : INITIAL (0),
```

```
355 0482 1 !! xntab : REF xntab_define, | List of transaction numbers assigned
356 0483 1 !! xpagen : REF xpagen_define, | Page number associated with transaction number.
357 0484 1 sca : sca_definition, | Scanner control area.
358 0485 1 scalit : VECTOR [sca_case_size], | Save case rules here during literal
359 0486 1 | processing.
360 0487 1 numprm : numprm_define, | Work area.
361 0488 1 fs_allocate (fs01, 150), | General purpose character string
362 0489 1 | work area.
363 0490 1 outopt : outopt_define, | Output options.
364 0491 1 frmstd, | Depth of FRMSTK (0 means empty).
365 0492 1 frmstk : form_stack,
366 0493 1 hlc : hlc_definition,
367 0494 1 hldsp : VECTOR [max_levels],
368 0495 1 lstchr : REF VECTOR,
369 0496 1 lstlch : REF VECTOR,
370 0497 1 lstrch : REF VECTOR,
371 0498 1 lstldd : REF VECTOR,
372 0499 1 lstcnt : REF COUNTED_LIST,
373 0500 1 lstskp : REF VECTOR,
374 0501 1 lchlst : VECTOR [nmlst_maxsize],
375 0502 1 rchlst : VECTOR [nmlst_maxsize],
376 0503 1 lddlst : VECTOR [nmlst_maxsize],
377 0504 1 chlst : VECTOR [nmlst_maxsize],
378 0505 1 nmlst : number_list,
379 0506 1 pdt : REF pdt_definition,
380 0507 1 phan : phan_definition,
381 0508 1 skplst : VECTOR [nmlst_maxsize],
382 0509 1 iobstk : BLOCKVECTOR [max_require, iob$k_length],
383 0510 1 |
384 0511 1 | Active input file IOBs saved in IOBSTK.
385 0512 1 |
386 0513 1 rneiob : REF $XPO_IOB (), | Storage for primary input IOB.
387 0514 1 rneiob : REF $XPO_IOB (), | Storage for output IOB.
388 0515 1 rneiob : REF $XPO_IOB (), | Input IOB currently active.
389 0516 1 rneiob : REF $XPO_IOB (), | Output file IOB
390 0517 1 | (i.e., for generated document).
391 0518 1 tteiob : $XPO_IOB (), | Standard error file (usually TTY:).
392 0519 1 ttiob : $XPO_IOB (), | Standard input file (usually TTY:).
393 0520 1 ttoiob : $XPO_IOB (), | Standard output file (usually TTY:).
394 0521 1 tsiob : $XPO_IOB (), | Standard input file; but STREAM,
395 0522 1 | not RECORD.
396 0523 1 |
397 0524 1 | Declare a string descriptor that will point to the "resultant" file
398 0525 1 | name string when any file operation fails.
399 0526 1 |
400 0527 1 ffname : $STR_DESCRIPTOR (CLASS = DYNAMIC),
401 0528 1 |
402 0529 1 | Declare a field to contain a secondary error code. This field will be
403 0530 1 | stuffed with a code explaining why the requested operation against the
404 0531 1 | file specified in FNAME failed. This error code will be defined by the
405 0532 1 | the I/O system being used (i.e. XPORT or RMS).
406 0533 1 |
407 0534 1 semcod,
408 0535 1 vrcnt, | Number of variables.
409 0536 1 vrname : vrname_def, | The variable names.
410 0537 1 vrlng : vrlng_def, | The length of the names.
411 0538 1 vrbool : vrbool_def, | TRUE/FALSE value.
```



```
412 0539 1      vrfflg      : vrfflg_def,      ! FALSE flag.
413 0540 1      vrtflg      : vrtflg_def,      ! TRUE flag.
414 0541 1      vrsrc       : vrsrc_def,       ! Source of variable definition.
415 0542 1
416 0543 1 %IF %BLISS (BLISS32) %THEN
417 0544 1      lffctx,      ! Global context cell used by LIB$FIND_FILE
418 0545 1 %FI
419 0546 1      ipftyp;      ! Will contain the index into the following table which
420 0547 1                      ! identifies what the input file's extension field is.
421 0548 1
422 0549 1 ! The following two lists of file types determine which output file type
423 0550 1 ! results from an input file type. The two lists must remain in order since
424 0551 1 ! translation is done by searching down one list and using the computed index
425 0552 1 ! to fetch the result from the other list. They are used in RUNOFF.BLI for
426 0553 1 ! the VMS implementation and CLH.BLI for the TOPS-20 implementation.
427 0554 1
428 0555 1      IPFTCT (above) must indicate the length of each table.
429 0556 1
430 0557 1 GLOBAL BIND
431 0558 1      ipftop = PLIT (      ! InPut File Type Options
432 0559 1
433 0560 1          CH$PTR(UPLIT('.RNO')), !Produces .MEM
434 U 0561 1 %IF DSRPLUS %THEN
435 U 0562 1          CH$PTR(UPLIT('.RTB')), !Produces .MET
436 0563 1 %FI
437 0564 1          CH$PTR(UPLIT('.RNT')), !Produces .MEC
438 0565 1          CH$PTR(UPLIT('.RNX')), !Produces .MEX
439 0566 1          CH$PTR(UPLIT('.RND')), !Produces .DOC
440 0567 1          CH$PTR(UPLIT('.RNH')), !Produces .HLP
441 0568 1          CH$PTR(UPLIT('.RNB')), !Produces .BLB
442 0569 1          CH$PTR(UPLIT('.RNC')), !Produces .CCO
443 0570 1          CH$PTR(UPLIT('.RNE')), !Produces .ERR
444 0571 1          CH$PTR(UPLIT('.RNL')), !Produces .PLM
445 0572 1          CH$PTR(UPLIT('.RNM')), !Produces .MAN
446 0573 1          CH$PTR(UPLIT('.RNP')), !Produces .OPR
447 0574 1          CH$PTR(UPLIT('.RNS')), !Produces .STD
448 U 0575 1 %IF DSRPLUS %THEN,
449 U 0576 1          CH$PTR(UPLIT('.*')) !Wild card file extension
450 0577 1 %FI
451 0578 1                      ): VECTOR,
452 0579 1
453 0580 1      opftop = PLIT (      ! OutPut File Type Options
454 0581 1
455 0582 1          CH$PTR(UPLIT('.MEM')), !From .RNO
456 U 0583 1 %IF DSRPLUS %THEN
457 U 0584 1          CH$PTR(UPLIT('.MET')), !From .RTB
458 0585 1 %FI
459 0586 1          CH$PTR(UPLIT('.MEC')), !From .RNT
460 0587 1          CH$PTR(UPLIT('.MEX')), !From .RNX
461 0588 1          CH$PTR(UPLIT('.DOC')), !From .RND
462 0589 1          CH$PTR(UPLIT('.HLP')), !From .RNH
463 0590 1          CH$PTR(UPLIT('.BLB')), !From .RNB
464 0591 1          CH$PTR(UPLIT('.CCO')), !From .RNC
465 0592 1          CH$PTR(UPLIT('.ERR')), !From .RNE
466 0593 1          CH$PTR(UPLIT('.PLM')), !From .PNL
467 0594 1          CH$PTR(UPLIT('.MAN')), !From .RNM
468 0595 1          CH$PTR(UPLIT('.OPR')), !From .RNP
```



```
: 469      0596 1      CH$PTR(UPLIT('STD')) !From .RNS
: 470      U 0597 1 %IF DSRPLUS %THEN,
: 471      U 0598 1      CH$PTR(UPLIT('MEM')) !From .***
: 472      0599 1 %FI
: 473      0600 1      ): VECTOR;
: 474      U 0601 1 %IF DSRPLUS %THEN
: 475      U 0602 1 !      Various keyword table listheads.
: 476      U 0603 1 GLOBAL
: 477      U 0604 1      kwtrpn,      ! .ReferencePoint Names
: 478      U 0605 1      kwtcbr,      ! Cross-Book crossReference structures
: 479      U 0606 1      kwtdm;      ! User Defined Macros (counter variable names)
: 480      U 0607 1
: 481      U 0608 1 $keyword_table (kwtpdm, G, ! PreDefined Macros ('$$' entity names)
: 482      U 0609 1      (APPENDIX,      kw$pdm_appendix),
: 483      U 0610 1      (CHAPTER,      kw$pdm_chapter),
: 484      U 0611 1      (CURRENT_FILE, kw$pdm_current_file),
: 485      U 0612 1      (DATE,      kw$pdm_date),
: 486      U 0613 1      (EXAMPLE,      kw$pdm_example),
: 487      U 0614 1      (FIGURE,      kw$pdm_figure),
: 488      U 0615 1      (FOOTNOTE,      kw$pdm_footnote),
: 489      U 0616 1      (HEADER_LEVEL, kw$pdm_header_level),
: 490      U 0617 1      (MARGIN,      kw$pdm_margin),
: 491      U 0618 1      (MARK,      kw$pdm_mark),
: 492      U 0619 1      (PAGE,      kw$pdm_page),
: 493      U 0620 1      (PAGE_SIZE,      kw$pdm_page_size),
: 494      U 0621 1      (ROOT_FILE,      kw$pdm_root_file),
: 495      U 0622 1      (RUNNING,      kw$pdm_running),
: 496      U 0623 1      (SECTION,      kw$pdm_section),
: 497      U 0624 1      (SPACING,      kw$pdm_spacing),
: 498      U 0625 1      (TABLE,      kw$pdm_table),
: 499      U 0626 1      (TIME,      kw$pdm_time)
: 500      U 0627 1      );
: 501      U 0628 1
: 502      U 0629 1 $keyword_table (kwtpdp, G, ! PreDefined Parameters ('$$' entity names)
: 503      U 0630 1      (DAY,      kw$pdp_day),
: 504      U 0631 1      (HOUR,      kw$pdp_hour),
: 505      U 0632 1      (LEFT,      kw$pdp_left),
: 506      U 0633 1      (LENGTH,      kw$pdp_length),
: 507      U 0634 1      (MINUTE,      kw$pdp_minute),
: 508      U 0635 1      (MONTH,      kw$pdp_month),
: 509      U 0636 1      (RIGHT,      kw$pdp_right),
: 510      U 0637 1      (SECOND,      kw$pdp_second),
: 511      U 0638 1      (WIDTH,      kw$pdp_width),
: 512      U 0639 1      (YEAR,      kw$pdp_year)
: 513      U 0640 1      );
: 514      U 0641 1 %FI
: 515      0642 1
: 516      0643 1 GLOBAL BIND      ! Flag table search order.
: 517      0644 1      fltso = PLIT (
: 518      0645 1
: 519      U 0646 1 %IF DSRPLUS %THEN
: 520      U 0647 1      NPX_FLAG,      ! <Nopermute Index flag>
: 521      0648 1 %FI
: 522      0649 1      SUB_FLAG,      ! <SUBSTITUTE flag>
: 523      0650 1      QUO_FLAG,      ! <QUOTE flag>
: 524      0651 1      UPP_FLAG,      ! <UPPERCASE flag>
: 525      0652 1      LOW_FLAG,      ! <LOWERCASE flag>
```



```
: 526      0653 1      CAP_FLAG,  
: 527      0654 1      UND_FLAG,  
: 528      0655 1      BLD_FLAG,  
: 529      0656 1      SPA_FLAG,  
: 530      0657 1      IND_FLAG,  
: 531      0658 1      HYP_FLAG,  
: 532      0659 1      BRK_FLAG,  
: 533      0660 1      PER_FLAG,  
: 534      0661 1      OVR_FLAG)  
: 535      0662 1      : VECTOR;  
: 536      0663 1  
: 537      0664 1 END  
: 538      0665 0 ELUDOM
```

! End of module.

```
.TITLE  GLBDAT Define global literals and data structur  
es  
.IDENT  \V04-000\  
.PSECT  $SPLITS,NOWRT,NOEXE,2
```

```
4F 4E 52 2E 00000 P.AAB: .ASCII \.RNO\  
54 4E 52 2E 00004 P.AAC: .ASCII \.RNT\  
58 4E 52 2E 00008 P.AAD: .ASCII \.RNX\  
44 4E 52 2E 0000C P.AAE: .ASCII \.RND\  
48 4E 52 2E 00010 P.AAF: .ASCII \.RNH\  
42 4E 52 2E 00014 P.AAG: .ASCII \.RNB\  
43 4E 52 2E 00018 P.AAH: .ASCII \.RNC\  
45 4E 52 2E 0001C P.AAI: .ASCII \.RNE\  
4C 4E 52 2E 00020 P.AAJ: .ASCII \.RNL\  
4D 4E 52 2E 00024 P.AAK: .ASCII \.RNM\  
50 4E 52 2E 00028 P.AAL: .ASCII \.RNP\  
53 4E 52 2E 0002C P.AAM: .ASCII \.RNS\  
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00030 .LONG 12  
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00034 P.AAA: .ADDRESS P.AAB, P.AAC, P.AAD, P.AAE, P.AAF, -  
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 0004C P.AAG, P.AAH, P.AAI, P.AAJ, P.AAK, P.AAL, -  
P.AAM  
4D 45 4D 2E 00064 P.AAO: .ASCII \.MEM\  
43 45 4D 2E 00068 P.AAP: .ASCII \.MEC\  
58 45 4D 2E 0006C P.AAQ: .ASCII \.MEX\  
43 4F 44 2E 00070 P.AAR: .ASCII \.DOC\  
50 4C 48 2E 00074 P.AAS: .ASCII \.HLP\  
42 4C 42 2E 00078 P.AAT: .ASCII \.BLB\  
4F 43 43 2E 0007C P.AAU: .ASCII \.CCO\  
52 52 45 2E 00080 P.AAV: .ASCII \.ERR\  
4D 4C 50 2E 00084 P.AAW: .ASCII \.PLM\  
4E 41 4D 2E 00088 P.AAX: .ASCII \.MAN\  
52 50 4F 2E 0008C P.AAY: .ASCII \.OPR\  
44 54 53 2E 00090 P.AAZ: .ASCII \.STD\  
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00094 .LONG 12  
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00098 P.AAN: .ADDRESS P.AAO, P.AAP, P.AAQ, P.AAR, P.AAS, -  
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 000BC P.AAT, P.AAU, P.AAV, P.AAW, P.AAX, P.AAY, -  
P.AAZ  
00000007 00000006 00000005 00000004 00000003 00000000 000C8 P.ABA: .LONG 13  
0000000F 00000010 0000000B 0000000A 00000009 00000008 000CC .LONG 0, 3, 4, 5, 6, 7, 8, 9, 10, 11, 16, 15, 12  
0000000C 000FC
```



```
                                .PSECT $OWNS$,NOEXE,2
00000 PP_SCA: .BLKB 48
                                .PSECT $GLOBALS$,NOEXE,2
00000 R_CMD:: .BLKB 100
00064 LSTSTF:: .BLKB 68
000A8 TSFA:: .BLKB 160
00148 MRAA:: .LONG 0, 0, 500, 0
00158      .BLKB 500
0034C FRA:: .LONG 0, 0, 500, 0
0035C      .BLKB 500
00550 IRA:: .LONG 0, 0, 10, 0
00560      .BLKB 10
0056A      .BLKB 2
0056C XMRA:: .LONG 0, 0, 100, 0
0057C      .BLKB 100
005E0 MRA:: .BLKB 4
005E4 TITTSF:: .BLKB 160
00684 SBTTSF:: .BLKB 160
00724 TITMRA:: .LONG 0, 0, 450, 0
00734      .BLKB 450
008F6      .BLKB 2
008F8 SBTMRA:: .LONG 0, 0, 450, 0
00908      .BLKB 450
00ACA      .BLKB 2
00ACC XTSF:: .BLKB 160
00B6C KHAR:: .BLKB 4
00B70 FLGT:: .BLKB 144
00C00 TTABLE:: .BLKB 168
00CA8 ATABLE:: .BLKB 256
00DA8 PDTSTF:: .BLKB 12
00DB4 SAVE:: .BLKB 80
00E04 SAVEALL:: .BLKB 44
00E30 DATTIM:: .BLKB 24
00E48 HLLIST:: .BLKB 32
00E68 ECC:: .BLKB 432
01018 SAVSTK:: .BLKB 876
01384 TSF:: .BLKB 4
01388 FNESIZ:: .BLKB 84
013DC FNISIZ:: .BLKB 84
01430 FNCT:: .BLKB 48
01460 GCA:: .BLKB 256
01560 HCT:: .BLKB 48
01590 IFSTK:: .BLKB 356
016F4 FOHLCH:: .BLKB 4
016F8 FOHRCH:: .BLKB 4
016FC FOHDD:: .BLKB 4
01700 FOHLSC:: .BLKB 4
01704 FOHLSK:: .BLKB 4
01708 FOHLST:: .BLKB 4
0170C FOHPDT:: .BLKB 4
01710 FOHMRA:: .BLKB 4
01714 FOHTSF:: .BLKB 4
```



```
00000000 000001F4 00000000 00000000 01718 FOOREC::.BLKB 20
0172C FOOMRA::.LONG 0, 0, 500, 0
0173C .BLKB 500
01930 FOOIOB::.BLKB 4
01934 FOTIOB::.BLKB 4
01938 FO1IOB::.BLKB 244
01A2C FO2IOB::.BLKB 244
01B20 FOOPDT::.BLKB 12
01B2C FOOSCA::.BLKB 384
01CAC F_LIST::.BLKB 20
01CC0 F_LCHR::.BLKB 12
01CCC F_LSKP::.BLKB 12
01CD8 F_LLCHR::.BLKB 12
01CE4 F_LRCHR::.BLKB 12
01CF0 F_LDD::.BLKB 12
01CFC FOOTSF::.BLKB 160
01D9C IRAC::.BLKB 28
01DB8 NPAGEN::.BLKB 16
01DC8 SPAGER::.BLKB 80
01E18 TPAGER::.BLKB 80
01E68 PAGEN::.BLKB 16
01E78 BRNOOB::.BLKB 244
01F6C BRNIOB::.BLKB 244
02060 SCA::.BLKB 384
021E0 SCALIT::.BLKB 100
02244 NUMPRM::.BLKB 16
00000000 00000096 00000000 00000000 02254 FS01::.LONG 0, 0, 150, 0
02264 .BLKB 150
022FA .BLKB 2
022FC OUTOPT::.BLKB 24
02314 FRMSTD::.BLKB 4
02318 FRMSTK::.BLKB 960
026D8 HLC::.BLKB 40
02700 HLDSP::.BLKB 24
02718 LSTCHR::.BLKB 4
0271C LSTLCH::.BLKB 4
02720 LSTRCH::.BLKB 4
02724 LSTLDD::.BLKB 4
02728 LSTCNT::.BLKB 4
0272C LSTSKP::.BLKB 4
02730 LCHLST::.BLKB 160
027D0 RCHLST::.BLKB 160
02870 LDDLST::.BLKB 160
02910 CHLST::.BLKB 160
029B0 NMLST::.BLKB 324
02AF4 PDT::.BLKB 4
02AF8 PHAN::.BLKB 76
02B44 SKPLST::.BLKB 160
02BE4 IOBSTK::.BLKB 2440
0356C RNEIOB::.BLKB 4
03570 RNAIOB::.BLKB 4
03574 RNIIOB::.BLKB 4
03578 RNOIOB::.BLKB 4
0357C TTEIOB::.BLKB 244
03670 TTIIOB::.BLKB 244
```

03764	TTOIOB::	.BLKB	244
03858	TSIOB::	.BLKB	244
0394C	FFNAME::	.BLKB	8
03954	SEMCOD::	.BLKB	4
03958	VRcnt::	.BLKB	4
0395C	VRNAME::	.BLKB	336
03AAC	VRLNG::	.BLKB	84
03B00	VRBOOL::	.BLKB	84
03B54	VRFFLG::	.BLKB	84
03BA8	VRTFLG::	.BLKB	84
03BFC	VRSRC::	.BLKB	84
03C50	LFFCTX::	.BLKB	4
03C54	IPFTYP::	.BLKB	4

RINTES==	28
S FMRA==	500
TMRMRA SIZE==	500
IPFTCT==	12
IPFTOP==	P.AAA
OPFTOP==	P.AAN
FLTSO==	P.ABA

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	48	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$GLOBALS	15448	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$SPLITS	256	NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
. ABS .	0	NOVEC, NOWRT, NORD, NOEXE, NOSHR, LCL, ABS, CON, NOPIC, ALIGN(0)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]XPORT.L32;1	590	85	14	252	00:00.1
\$255\$DUA28:[RUNOFF.SRC]DSRLIB.L32;1	1248	238	19	86	00:00.3

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:GLBDAT/OBJ=OBJ\$:GLBDAT MSRC\$:GLBDAT/UPDATE=(ENHS:GLBDAT)

Size:	0 code + 15752 data bytes
Run Time:	00:11.7
Elapsed Time:	00:26.2

GLBDAT
V04-000

Define global literals and data structures
Global storage

F 7
16-Sep-1984 00:41:32

VAX-11 Bliss-32 V4.0-742

Page 15

: Lines/CPU Min: 3416
: Lexemes/CPU-Min: 44583
: Memory Used: 135 pages
: Compilation Complete

GNA
V04

0342

AH-BT13A-SE
 VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY